Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

ALSP - THEORY, EXPERIENCE, AND FUTURE DIRECTIONS

Richard M. Weatherly
Annette L. Wilson
Sean P. Griffin

The MITRE Corporation
7525 Colshire Drive
McLean, Virginia 22102, U.S.A.

ABSTRACT

The Aggregate Level Simulation Protocol (ALSP) interface technique is now entering its second year of practical application. ALSP is designed to permit multiple, pre-existing combat simulations to interact with each other over local and wide area networks. Army, Navy, and Air Force simulations have been linked, using ALSP, to form a confederation of models capable of supporting major military exercises. This exercise experience has reinforced many of the initial ALSP design decisions, called others into question, and revealed some new requirements. The basic ALSP system software services, practical features derived of operational experience, and the current research efforts are described.

1 TECHNICAL APPROACH

ALSP assumes a logical architecture of its components that can be viewed as shown in Figure 1. This grouping is called an ALSP confederation. This is a logical view and does not consider which computers might host which components, or the physical path for communications between the components.

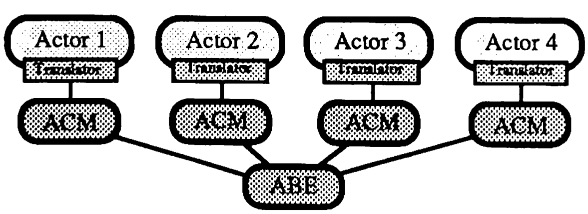

Figure 1: ALSP Logical Architecture

1.1 Actor

An actor typically is a warfare simulation. Most existed before integration using ALSP was considered. Most must still operate alone in other exercises. Some actors operate in peripheral roles—global displays, event loggers, etc. In all cases, it is a part of the ALSP philosophy to modify the operation (computer code) of the actor so that it can operate in an ALSP confederation, but to minimize the extent of the modifications.

Three major modifications to an actor are required. The first is to recognize that some simulation objects in its game space are not its to control. This requires special logic to prevent manipulation of foreign or "ghosted" objects.

The second modification is to recognize its responsibility to inform the rest of the confederation of significant changes to simulation objects that it controls. This responsibility requires that changes to significant attributes, such as location, be reported. It also requires that significant events between controlled objects and ghosted objects, like combat, be reported.

The third modification requires an actor to coordinate its advance of simulation time with the confederation. While the coordination logic resides in the translator, very often the mechanism of time advance in the actor must be modified.

1.2 Translator

A translator is a set of computer code that provides the bulk of the modifications to an actor. It is segregated from the actor code but linked with it to produce an integrated process. While some functions provided by a translator are common to all translators (communicating with the ACM for example), most functions are unique and specific to the actor that the translator supports.

1.3 ALSP Common Module (ACM)

An ACM is the glue that holds the ALSP confederation together. Only one version of the ACM functions for all actor/translators, although a separate copy is provided for each as shown in Figure 1. This mechanism provides a common point of interface for all translators. It also permits actors to join a confederation without knowing or caring who the other members are.

Services provided by the ACM include:

- Coordination of the process of joining and departing from the ALSP confederation.
- Coordination of actor local time with confederation time.
- Filtering of incoming messages so that only those of interest are received by the translator.
- Coordination of the ownership of attributes of objects so that this ownership can migrate between actors.
- Enforcement of the ownership of attributes so that actors may only report new values for attributes of objects that they own.

1.4 ALSP Broadcast Emulator (ABE)

An ABE is a process that facilitates the distribution of ALSP information. Its primary function is to receive a message on one of its communications paths and retransmit the message on all of its remaining communications paths. This permits configurations like that depicted in Figure 1 where all ALSP components are local to one another (in the same computer or on a local area network). It also permits configurations where sets of ACMs communicates with their own ABE and the ABEs communicate among one another over wide area networks, for example.

2 HISTORY OF ALSP

The ALSP concept formulation evolved through analysis of existing warfare simulations. Based on experiences with aggregate level warfare simulations in use at the Warrior Preparation Center (WPC), MITRE personnel proposed that ALSP be investigated, as an ARPA funded prototyping effort, with the aim of generalizing and systematizing the simulation interface process. ARPA guidance in this investigation was to use as many of the SIMNET, now DIS, principles as possible in the design of ALSP. Principles central to DIS that were used in ALSP are:

- No Central Node - simulators joined and departed from the confederation at will
- Geographic Distribution - simulators could be distributed to different geographic locations yet exercise over the same terrain
- Object Ownership - each simulator controlled its own resources, firing its own weapons and determining the appropriate damage to its systems when fired upon
- Message-based Protocol - information from one simulator was distributed to all other simulators using a defined message-based protocol.

Additional requirements of aggregate-level simulations necessary that were not a primary concern in DIS are:

- Time Management - Simulation times must be coordinated so that the times for all simulations would appear the same to users and so that event causality would be maintained. The goal was to have events occur in the same sequence in all simulations.
- Data Management - A method was needed to permit all simulations to share information in a commonly understood manner given each had its own representation of data. This included the need for multiple simulations to control attributes of the same object.
- Architecture - A method was needed to permit the simulations to continue to use their existing architectures, yet exist in an ALSP confederation.

In January 1991, a ground combat simulation, GRWSIM, was selected for use in determining if the ALSP concepts could function in the real warfare simulation world. Analysis confirmed that the GRWSIM code could be modified to participate in a confederation of models. A set of reusable code was developed in Ada to facilitate the integration. Ada was selected because this computer language provides features that ease the development of the required software and because it has been specified as the standard DOD development language. The Chandy-Misra algorithm was selected to provide a conservative time coordination method.

Once the experiments with GRWSIM were completed using reusable ALSP software, the success prompted the next step: a GRWSIM-AWSIM demonstration which occurred in June 1991. The software was again obtained from WPC and modified with reusable code. Success with the joining of these simulations led to the decision to link AWSIM and CBS.

In September 1991, an AWSIM-CBS demonstration took place in the MITRE laboratory as a join effort with JPL. Experience with the successful AWSIM-CBS lab demonstrations resulted in a decision to use ALSP in support of a major exercise – Reforger 92.

The decision forced a compressed schedule for development of exercise-quality software. The effort was further complicated by realization that the lab architecture for ALSP, where the ALSP software attached to the simulation, was not generally usable. A new architecture was devised, where the ALSP software would reside in a process separate from the simulation and would communicate with it using a message-based protocol. This new architecture caused much of the laboratory computer code to be obsolete.

A closely coordinated development process began in September 1991, resulting in a new ALSP architecture and new translators (the name for code added to a simulation that interacts with ALSP) for AWSIM and CBS. In keeping with an ALSP philosophy that warfare simulations are best modified by agencies responsible for maintaining them, three agencies were involved in the development process: MITRE, LANL, and JPL. JPL was responsible for the CBS development with LANL responsible for the AWSIM development. MITRE was responsible for ALSP software development which included the ACM and ABE.

In addition to software development, the simulation-to-simulation portions of the protocol required refinement. Representatives of the developing and using agencies undertook this task over several working sessions. The result of this effort was a CBS-AWSIM Interface Control Document (ICD) produced by JPL.

In preparation for Reforger 92, ALSP was incorporated into several exercises scheduled for earlier in the year, Central Fortress 92 and Ulchi Focus Lens 92 which were held in June and August 1992, respectively. Each exercise offered the opportunity to challenge ALSP's capabilities and expand its functionality. For example, in Focus Lens ALSP ran over a wide area network between sites in Germany and Korea without major problems.

The Reforger 92 exercise was a US Army, Europe (USAREUR) exercise that was conducted in Germany in September and October 1992. In this exercise, both CBS and AWSIM were executed at WPC. ALSP performed as expected and resulted in two major decisions:

1. ALSP moved from being a ARPA prototype project to an ongoing STRICOM project; and

2. the Research, Evaluation and Systems Analysis (RESA) model; the US Army intelligence simulation (TACSIM); and the Joint Electronic Warfare Simulation (JECEWSI) began efforts to join the AWSIM-CBS confederation.

In June 1993, a confederation of CBS-RESA-AWSIM was successfully tested at the WPC in preparation for Ulchi Focus Lens '93. This exercise is scheduled for August 1993 and will be the first ALSP confederation to contain major simulations from the Army, Navy, and Air Force.

3 LESSONS LEARNED

The first year's experience with ALSP produced two surprises:

1. A large amount of interface functionality was created with a very manageable demand in communication services. The current ABE based design sends all messages to all ACMs. The ACM then filters the message stream based on criteria proved by the actor. While this approach has been sufficient, section 4 below describes work intended to reduce further the demand for communications services and accommodate larger confederations.

2. Management of an ALSP confederation involves much more than simply assuring that the ALSP system software is operating correctly. Any interface system that is expected to serve the training community must take into account the nature of that human organization.

3.1 Management Tools

A facility has been added to the ALSP system software to permit each ACM and ABE to be controlled from as many as four separate locations. This requirement came from practical exercise experience. During an exercise, a confederation manager position is usually established where all ALSP components receive their primary control. Second, each model's operators observe the state of their particular ACM. Third, the overall exercise manager monitors all ALSP components for general information and as a backup to the confederation manager. In addition to these practical requirements, major exercises attract large numbers of visitors who would like to see the software in action. The need for a visitor's demonstration area usually accounts for the fourth set of ALSP monitors.

These requirements for visibility into the ALSP system software have prompted the development three different operational views.

- ACM Monitor Screen - Displays detailed information about the state of the ACM internal processes, the Chandy-Misra queues, and other status and performance information required to diagnose and repair faults.

- Actor Monitor Screen - Indicates the ACM's perception of the actor's state. This includes the number of ghosted and owned objects tabulated by object class and information about the actor's progress in both simulation and real time.

- Confederation Monitor Screen - Tabulates each Actor, ACM and ABE in the confederation. For each actor, the status of its communication channels, its ratio of simulation time to real time, and other high level information is displayed. This screen provides assurance at a glance that the confederation is advancing normally and should there be a problem, high level guidance on how to correct it.

4 THE EVENT DISTRIBUTION PROTOCOL

The current research component of the ALSP project addresses growth questions associated with the assumptions of reliable, order preserving multicast provided by the ABE. It is clear that as the number of actors in the confederation increases, the strategy of sending all information to all actors will fail in the face of an N^2 growth in demand for communications and processing. To address this, MITRE is developing the Event Distribution Protocol (EDP) to intelligently distribute object state change information between actors. The EDP project includes extensions to the peer-level protocol used between ACMs, new software to replace the ABE, and statistical study of message traffic in current ALSP confederations. To determine where opportunities for traffic reduction exist, the method by which actors become aware of objects owned by other actors in an ALSP confederation is reviewed.

4.1 The Object Discovery Process

Ideally, the generator of an update message would be able to determine the exact destination and content of each message it produces given knowledge of all the other actor's requirements for information. Such an approach leads to an unacceptable growth in replicated object information among the ACMs because each ACM must make the object discovery decisions for all the other ACM's in the confederation each time one of its objects changes state. The object discovery process and the information required to compute it is outlined below.

When an ACM receives an update message from the confederation it asks two questions:

1. Is the object currently known (being ghosted) by my client actor?
2. Does this update pass my client actor's filter criteria:
   a. Is the updated object an instance of an object class which the client actor is interested in?
   b. Given the object's class is desirable, does the update contain information about object attributes of concern to the actor?
   c. Within the list of desirable object attributes, do the updated attribute values satisfy all specified range constrains placed by the actor?

The result of these questions is applied to the decision square in Figure 2.

Passes Filter Criteria	Object Known: Yes	Object Known: No
Yes	Update	Create
No	Delete	Discard

Figure 2: The Object Discovery Decision

The actions taken as a result of the object discovery decision are explained below:

- Update - Pass the newly received attribute values to the actor so that it can modify its perception of the object.
- Create - Inform the actor of the existence of a new object and provide it with an initial set of attribute values.
- Delete - Instruct the actor to stop including this object in its perception of the combat environment.
- Discard - Extinguish the update message and send nothing to the actor.

4.2 Object Class and Attribute Based Reduction

If the cost of replicating all the information needed to distribute the discovery process across the confederation is too high, then what can be done to reduce the flow of undesired messages between ACMs? The compromise taken in the EDP is to distribute part of the filter criteria among the ACMs so they can apply it to reduce the amount of information they produce but defer the discovery decision to the recipient. If the ABEs are also modified to restrict message flow based on a knowledge of the recipient's needed, most of the "Discard" cases in the discovery decision can be eliminated.

It is important to note that only parts 2a and 2b of the filter criteria question above can be distributed. The class of an object and the set of attributes for that class are static for the life of the object. Thus, the discovery decision from a filter criteria perspective is only a function of attribute values.

Figure 3 depicts a four ACM confederation with two ABEs joined by a tie link. Each arc is labeled with the partial criteria that restricts the messages that flow along it. Criteria C_i is the filter criteria received by ACM_i from its client actor.

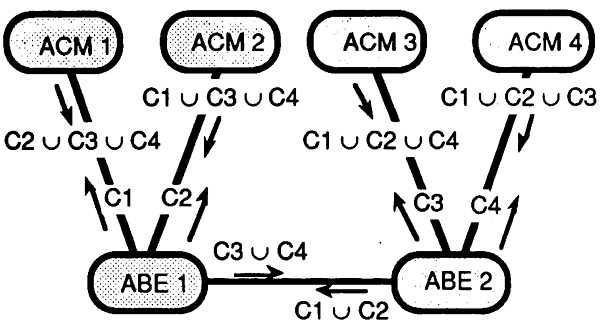

Figure 3: Filter Criteria Generated by EDP

5 FUTURE DIRECTIONS

Two paths are being pursued to incorporate the EDP results into the production ALSP system:

- Statistical validation of the EDP potential.
- ACM modifications to accommodate the reception and generation of filter criteria.

In June 1993, an AWSIM-CBS-RESA-TACSIM functional validation test was held at the WPC. This test furnished an opportunity to collect fairly realistic data without the danger of disturbing an actual exercise. Assuming continued positive results from the analysis of this data, an EDP based ALSP system will be available in time for the 1994 exercise season. Results of the data analysis and experience with an EDP based system will be the subject of a future publication.

ACKNOWLEDGMENTS

This work was partially supported by a MITRE Mission Oriented Investigation and Exploration (MOIE) project under contract number DAAB07-93-C-N651.

REFERENCES

Chandy, K.M., J. Misra. 1979. "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs." *IEEE Transactions on Software Engineering*, vol. SE-5, no. 5 (September).

AUTHOR BIOGRAPHIES

RICHARD M. WEATHERLY is a Principal Engineer for the MITRE Corporation in their C^3 Modeling and Simulation Center. He received his Ph.D. in Electrical Engineering from Clemson University in 1984. He is an original member of the ALSP team and the designer of the ALSP system software.

ANNETTE L. WILSON is a Member of the Technical Staff for the MITRE Corporation. She received her B.S. in computer science *Magna Cum Laude* from Texas A&M in 1986. She participated in the design and development of the production ALSP and EDP software systems.

SEAN P. GRIFFIN is a Member of the Technical Staff in the C^3 Modeling and Simulation Center at the MITRE Corporation. He received a B.S. degree in Applied Mathematics from the University of Rochester in 1991, and he is currently pursuing a M.S. degree in Information Systems at George Mason University.